INF 553 – Spring 2018

Homework #1: MapReduce & Spark

Due: February 5, Monday 11:59 pm PT
100 points

In this homework, we consider analyzing a dataset, menu.csv, that contains a set of (historic) menus of restaurants. The data set is a CSV file, containing id, name, sponsor, event, date, page_count, etc. Note that the data is noisy. For example, the "event" column may contain "DINNER", "[DINNER]", "DINNER;" which all mean that the menu was for a dinner event. Page_count is the number of pages the menu has. More information about the data set can be found here: http://menus.nypl.org/data

Note that dataset contains header information which you need to skip in your program.

Task 1: [Hadoop MapReduce, 60 points]

Consider the event and page_count columns of the data set. Write a Hadoop MapReduce program (in Java, see WordCount.java for example) **FirstName_LastName_Average.java.** For each unique event in the event column, the program computes the number of menus (Rows) that contain the event and the **average** number of pages of these menus. You may ignore menus whose events are empty.

Two events are considered to be identical if they contain the same set of tokens after the following transformations.

- Replace all punctuation characters (except for apostrophe ' and dash -) in the event value with white spaces. For apostrophe and dash, you simply remove it (without replacing it with space). Remove leading and trailing white spaces if any.
- Tokenize the event value using whitespaces as delimiters. If there are continuous multiple white spaces within the text, consider them as a single space and then tokenize.
- Lower-case the tokens.

For example, "NEW YEAR'S DINNER" is the same as "NEW YEARS DINNER". "LINCOLN'S BIRTHDAY DINNER" is the same as "DINNER, LINCOLNS BIRTHDAY". "LUNCH AND DINNER" is the same as "dinner and lunch". ABEND-ESSEN is the same as ABENDESSEN. [?REUNION?] is same as "reunion".

Example execution:

bin/hadoop jar FirstName_LastName_avg.jar FirstName_LastName_Average input-dir output-dir

where FirstName_LastName_avg.jar is the jar file created for your class, input-dir contains a single file: menu.csv and output-dir is where output (part-nnnnn) gets created .

Each line in the output contains the event, number of rows that contain the event and an average page count separated by tab. Lines are sorted in the ascending order of event values. The event values should be in lowercase.

Example output (Not an actual output)

breakfast 940 3.5
breakfast and dinner 490 4
dinner 180 5
…

Task 2: [Spark, 40 points]

Solve the same problem as above, but use Apache Spark for Python. Name your program **FirstName_LastName_Average.py**. Output should be written to a single file and should be created in the output-dir specified in the command line.

Example execution:

bin/spark-submit FirstName_LastName_Average.py menu.csv output-dir

Submission on Blackboard:

A zip file **FirstName_LastName_hw1.zip** containing the below:

a. A readme file that contains instructions to run your programs for task 1 and task 2 in the example execution format specified above. Please specify the Python and Spark version you are using.
b. FirstName_LastName_Average.java
c. FirstName_LastName_avg.jar
d. FirstName_LastName_task1.txt that contains the output from task 1
e. FirstName_LastName_Average.py.
f. FirstName_LastName_task2.txt that contains the output from task 2

Grading Criteria:

a. If java file is not submitted for task 1, there is 20% penalty.
b. If program does not run according to your Readme file, there is 50% penalty. In that case, grading will be done based on your output file submitted.
c. If output is not sorted, there is 20% penalty.
d. If more than one file is created in the output, there is 20% penalty.
e. Assignment is due at 11:59 pm on 02/05. Late homework will have 10% of penalty for every 24 hours that it is late. No credit will be given after 72 hours of its due time.

Important Note: All the submitted work must be your own. Do not share the code with anyone! For effective learning, start early!